

# **Basic Profiling With MPI**

**Luke Lonergan**  
**High Performance Technologies, Inc.**

# Basic MPI Profiling

After this exercise, we will have graphically profiled the 1-D Decomposed MPI code we developed in the previous exercises, and will have a good feel of how the communications overhead relates to program performance. This is fundamental to understanding how well the code will run on different machines.

In the last sections we developed a Stommel model in FORTRAN, and decomposed it in 1-D. We learned how to develop basic parallel Finite Difference methods using MPI.

# Use of MPICH MPE: mpilog

Change directory to mpi\_1d\_profiling/step2

Edit the README file

Recommended Approach:

Step through the exercise in the README file  
(and the FORTRAN)

View the balance of computation and  
communication graphically and change program  
size and number of processors to get a feel for the  
system (same as for the last exercise)

# Use of MPICH MPE: mpilog

Example:

```
stml_mpi1d.x
```

```
vi stml_mpi1d.x_profile.log
```

```
:g/^$/d
```

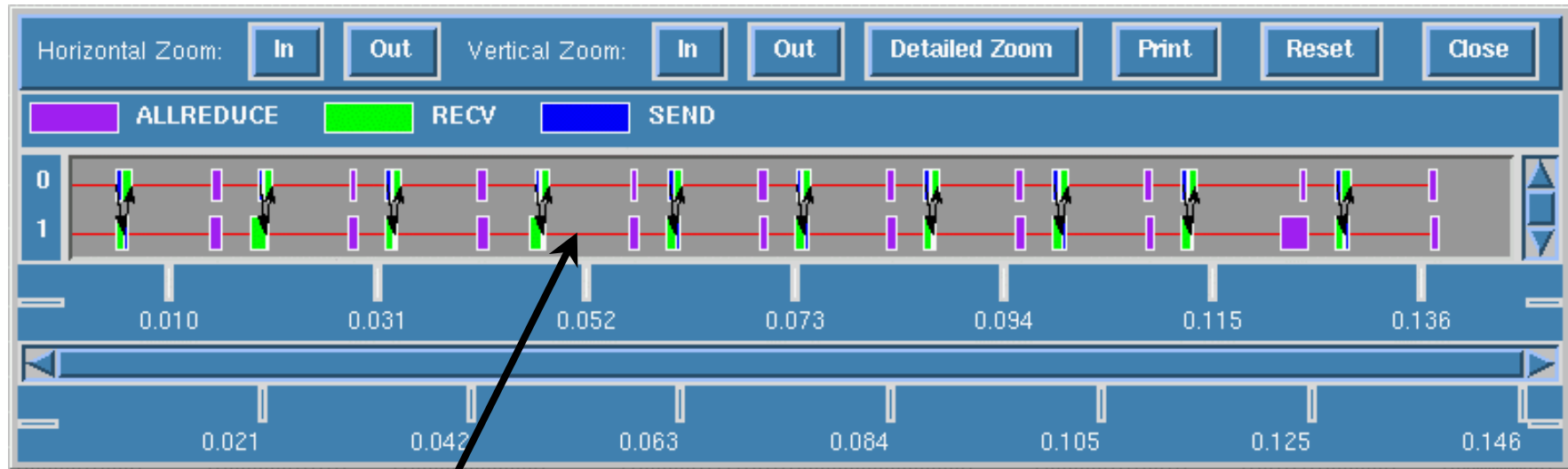
```
:wq
```

```
upshot stml_mpi1d.x_profile.log
```



Select to process the logfile

# Use of MPICH MPE: mpilog



**Red Timeline Uncovered by Communication Overhead is  
“clean” Execution**

Pay close attention to the ALLREDUCE  
communication as you increase CPUs! Can you  
improve it?

Duration: 30 Minutes